

17/PR 15

09/869980

JG17 d PCT/PTO 0 JUL 2001

- 1 -

SYSTEM AND METHOD FOR THREE DIMENSIONAL DATA REPRESENTATION

TECHNICAL FIELD

The present invention is directed toward presenting data in a useful
5 fashion, and, more specifically, to a system and method for determining how to display
data as data objects in an animated three-dimensional landscape, the presentation of
those objects, and the interaction between the data requester and the presented objects.

BACKGROUND OF THE INVENTION

Three-dimensional projections offer the viewer a "life-like"
10 representation that simulates what the viewer would see if the viewer was actually a part
of the projection. The three-dimensional projection need not be a projection of a real
place, but rather any landscape, real or imagined can be created. For instance, it would
be relatively easy to create a three-dimensional projection of a downtown area of an
actual city, or to create a virtual city that does not really exist.

15 Video games are an example of a case where a purpose of the three-
dimensional projection is to simulate a real-life environment. In these cases, the
purpose is to ensure to the user an immersion in a three-dimensional animated virtual
space and to provide to the user navigation functions that allow the user to explore the
three-dimensional virtual space that the video game presents.

20 Current computer technology provides a platform to easily create
realistic animated three-dimensional projections, such as those used for video games.
One popular type video game is a flight simulator program, which has been popular
since the beginning of the personal computer. Constant developments in computer
hardware and software allow the present generation of personal computers to run flight
25 simulation programs that nearly perfectly simulate flying a real plane. Fantasy games,
such as "Tomb Raider" by 3DO, invite the player to interact with the three-dimensional
world displayed on the computer screen. Role playing games such as Everquest and
Asheron's Call allow the user to interact with thousands of others in virtual worlds
connected by the Internet. The popularity of three-dimensional computer games is a

direct result of the ability of humans to comfortably interact in an simulated environment that is similar to the one we actually live in.

5 In these three-dimensional computer games, and in other similar cases, the user does not generally know much or any of the data used to create the three-dimensional space. The virtual topology is the data itself, and the topology does not correspond to any data know by the user. The animation and movement in these virtual spaces are only ways to interact with the virtual space, and that in itself is the purpose of playing the game.

10 Conversely, in other data presentation systems, the user is generally interested in the actual data presented. Conventionally, one or two-dimensional presentations are used. For instance, lists of data are common, such as the result from a data search or other query. Also common are two-dimensional data presentations, for example spreadsheets, or the row-column reports shown in Figures 1A and 1B. Typically, volumes of data are stored in a data repository, such as a database. Then, by
15 querying the database, desired data can be extracted therefrom. An example of extracted data is presented in Figure 1A. Before presenting this data to the user, in a form as shown in Figure 1B, some additional presentation data is added to the extracted data. For example, column titles identifying the data, such as "Identification", and "First field", etc. are added above the columns, while row labels are sometimes just the
20 data in the first column itself. For instance as seen in Figure 1B, the Identification numbers 001, 002, 003, 004 allow the user to easily locate a first, second and last field for each of the Identification numbers listed.

Data presented in such a one or two column form can be interacted with in the form of scrolling the rows or columns (if not presented in a one page view). The
25 interpretation of the data, i.e., its meaning, context, quality and all other characteristics must be deduced by the user interpreting the data.

To increase comprehension of numeric data, some forms of data visualization are available. For example, graphs and charts can help compare one set of data with one another. Also, data visualization techniques are known, such as those that
30 plot data into an x-y coordinate system. Furthermore, the same principles of data visualization can be applied to map data into a 3-dimensional space. For instance, Xerox PARC developed a method of data visualization through 3D mapping in CAM

Tree, as described in "An Easier Interface", by M. Clarkson in Byte Magazine (Feb., 1991). Additional visualization techniques are described in "The Ultimate User Interface", by Bob Jacobson, Byte Magazine (April, 1992).

5 In these and other similar techniques, the presentations create graphical effects to emphasize the quantitative meaning of data and the relationships between data. In these cases, the user knows the data, but the resulting presentation ensures a 3-dimensional view of the data.

10 Although the techniques known in the prior art allow data to be better visualized and understood than simply reviewing raw data, better techniques for immediately communicating data to viewers are still needed.

SUMMARY OF THE INVENTION

15 Embodiments of the invention provide a method and system to present data resulting from an inquiry in a comprehensive manner, where the user does not need to read text to completely understand the data retrieved, but rather need only look at a three-dimensional animated landscape containing the complete answer in graphical form. From the user's data query, a virtual world is created and presented to the user that appropriately simulates the requested data in virtual objects in a simulated realistic form.

20 Presented, therefore is a computer system and method for communicating results of a data query made to a data repository. After the data results from the data query are received from a database, factors that made up the query and the results of the query are matched to a set of presentation parameters that are stored in a second data repository. These parameters are combined back with the results of the data query to develop a listing of objects, which is further converted to a three-
25 dimensional landscape including virtual objects. In some embodiments the three-dimensional landscape is presented to a user on a viewing device. In other embodiments, the user can then interact with the three-dimensional landscape by an input device. In yet other embodiments, interaction with the three-dimensional landscape brings additional results from the original data query into the three-
30 dimensional view.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1A is an example of data presented in row-column form according to the prior art.

Figure 1B is an example of data presented in row-column form according
5 to the prior art.

Figure 2 is a functional block diagram showing example components of a three-dimensional data representation system according to an embodiment of the invention.

Figure 3 is a functional block diagram showing example components of a
10 three-dimensional data representation system according to another embodiment of the invention.

Figure 4 is a diagram showing results of a database query and the corresponding landscape produced by the three-dimensional data representation system of Figure 2 or 3.

Figure 5 is a functional diagram showing how the data representation
15 system of Figure 2 selects a particular landscape.

Figure 6 is a flowchart showing example main steps used to generate a landscape for a user based on a database query.

Figure 7 is a flowchart showing example main steps used to receive and
20 resolve a database query.

Figure 8 is a diagram showing detailed example steps for one of the steps of Figure 7.

Figure 9 is a flowchart showing example main steps used to construct an appropriate landscape.

Figure 10 is a diagram showing interrelationships between a repository
25 of virtual landscape parameters and other tables and landscapes.

Figure 11 is a flowchart showing example main steps used to display a virtual landscape.

Figure 12 is a diagram showing differences in granularity in two aspect
30 parameter tables.

Figures 13 – 16 are example diagrams showing examples of how a query to a database can be returned in a three dimensional graphic form that can be easily interpreted.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

5 The invention concerns a method for transforming data resulting from an inquiry performed by a user into a useful result for the user. It does this by transforming the lists of data produced by the user's query into an animated landscape simulating virtual three-dimensional space. Additionally, the user may interact with the three-dimensional landscape, by exploring, scrolling, and selecting presented objects.
10 By performing this action, the user is effectively interacting with the data produced by the query.

 This method synthesizes the three-dimensional image of the virtual space presented to the user by using characteristics of the actual search result sought by the user. In such a manner, the user feels a strong relationship between the formulated
15 query and the resulting three-dimensional landscape presented. For example, if the user is performing a query on hotels in a given city, the three dimensional landscape may be of an arbitrary street having a number of made-up hotels, corresponding one-to-one with the results of the user's query. For instance, if the data query produced two hotels that matched the query, two hotels would be presented in the three-dimensional
20 landscape shown to the user. If instead, the data query produced 30 hotels, then as many hotels as could effectively be placed on the presented landscape would be so placed, and the user could further interact with the non-represented data produced by the query (those hotels that could not fit on the screen) by interacting directly with the landscape, for example by "walking" down the street using three-dimensional
25 navigational tools to bring the other hotels within view.

 In this way, aspects of the virtual space and of the objects presented within the space are customized depending on, for example, the characteristics of the user, the type of data requested, the selection criteria, and the contents of the data produced by the query, among others. Therefore, because the search results are
30 presented in an environment with which the user is immediately comfortable, this

presentation allows a useful dialog between the user and the database, by using natural movement and exploration functions of a virtual three-dimensional world.

Figure 2 is a functional block diagram showing example components of a three-dimensional data representation system 200. Included within the data representation system 200 are a data retrieval system 210 that is coupled to a query request interpreter 220. The query request interpreter 220 interacts with a user interface 230, in that it accepts commands including requests for data and formulates them into a query that can be understood by the data retrieval system 210, and then forwards the query to the data retrieval system.

Results from the query are returned back to the query request interpreter 220, which are then sent to an aspect construction facilitator 240. This facilitator 240 reviews the results of the query, and interacts with a repository of virtual landscape aspect parameters 250 to determine which display parameters are best to report the results of the data query back to the user. Once the best display parameters are selected, the aspect construction facilitator 240 sends the parameters, and possibly the data results of the query themselves, to a display preparation facility 260. This facility 260 formats the results of the query, using the parameters retrieved by the aspect construction facilitator 240, and sends the formatted results back to the user interface 230, where they are rendered as a three-dimensional landscape to the user 270 that originally requested the data.

Figure 3 is a functional block diagram illustrating components of the three-dimensional data representation system 200 as implemented on a client-server system. In this embodiment, a server 310 would include the data retrieval system 210, the repository of virtual landscape parameters 250, and the aspect construction facilitator 240. Similarly, a client system 320 includes the query request interpreter 220, the display preparation facility 260, and the user interface 230. Communication between the client 320 and the server 310 is through a pair of data communicators, for example a server data communicator 330 and a client data communicator 332. These data communicators 330, 332 share all of the necessary information between the client 320 and the server 310. For instance, instead of the query request interpreter 220 directly sending the query to the data retrieval system 210, as was the case in the integrated system of Figure 2, it instead forwards this information to the data

communicator 332, which in turn sends the request to the data communicator 330. Once received, the data communicator 330 routes the request appropriately to the data retrieval system 210, and waits for a response. Once the response is received from both the data retrieval system 210 and the repository of virtual landscape aspect parameters 250, the aspect construction facilitator sends its response back to the data communicator 332, to be sent and used by the client system to formulate the three-dimensional landscape for the user 270.

These data communicators 330, 332 could communicate over a data connection 340, which could be any form that allowed effective data transfer between the client 320 and the server 310. Examples of possible data connections 340 include TCP/IP, PPP, or ATM connection, running over an ethernet, modem line, DSL link or cable modem, or any other appropriate communication protocol.

Figure 4 is a diagram showing results of a database query, and the corresponding landscape produced by the three-dimensional data representation system 200 of Figures 2 and 3. When referencing the data representation system 200, this description will not distinguish between the different implementations shown in Figures 2 and 3 because their functions are similar or identical even if their implementation is not.

In Figure 4, a user 270 has already entered a database query into the user interface 230, and it has been processed by the query request interpreter 220 which has received the results back from the data retrieval system 210. The results are shown in Figure 4 as a two-dimensional list of data 410, but the actual data generated by the query would not normally be shown to the user 270, and is only used as an example. The list of data 410 shows many records, with one record to each horizontal line. As mentioned above, oftentimes the first column in each record is used as its identification, which generally should be unique to the list of data 410. Additionally, each record has an entry in several fields, Field 1, Field 2, etc.

Each data element, i.e., the intersection of a particular row and column in the list of data 410, is represented, to the extent possible, in a landscape 420. In the landscape 420, the general scene is one of a street with many buildings located on either side.

Figure 5 is a diagram showing how the data representation system 200 of Figure 2 selects a particular landscape (such as the landscape 420 in Figure 4) for presentation to the user 270. To generate this landscape 420, the query and query results that resulted in the list of data 410 were sent by the query request interpreter 220 (Figure 2) to the aspect construction facilitator 240, which, in turn, sent a request to the virtual landscape parameter repository 250. The virtual landscape parameter repository 250 replied to the request by sending the aspect parameters for the request made by the aspect construction facilitator 240. The aspect construction facilitator 240, based on what was provided to it from the query results from the data retrieval system 210 and the virtual landscape parameter database 250, and additionally from a set of aspect recognition factors 510 that may be stored locally within the aspect construction facilitator, chose to instruct the display preparation facility 260 to prepare the landscape 420, rather than the landscapes 430 or 440. Depending on the combination of all of the variables received, the aspect construction facilitator 240 could have also selected either landscapes 430 or 440, or any other landscapes at its disposal. Additionally, the aspect construction facilitator 240 also instructed the display preparation facility 260 to populate the selected landscape with the data retrieved by the database query. For example, with reference to Figure 4, each of the Id's 001, 002 and 003 can be seen in the landscape 420 as individual buildings. The data stored in the Field 1 column is similarly shown as a large store window within each of the buildings. Note how the data from the list of data 410 matches with its respective row, such that Data 001 has the store window A associated with it, and store window B is associated with Data 002, etc. Further, the data from Field 2 is seen as one of the smaller windows in each building, again, matched to its respective building.

Some of the buildings, such as those representing the data 004 and 999 cannot be clearly seen in Figure-4. Embodiments of the invention also provide the user 270 with a set of tools enabling the user to navigate through the three-dimensional landscape 420. In doing so, for example, if the user 270 directed a virtual character (may or not be shown in the landscape) to walk down the street, the buildings 001 and 003 would go out of range and no longer be seen, while the buildings representing the data 004 and 999 would eventually be seen by the user 270. This is the equivalent of "scrolling" through a data list that is too large to be shown on a single screen.

Figure 6 is a flowchart showing example main steps used to generate a landscape for a user based on a database query. Generally speaking, step 610 receives and resolves the database query, step 640 determines which aspect parameters to use and begins construction of a three-dimensional landscape, and step 670 completes the three-dimensional landscape construction, including populating the landscape in whatever way best fits the data returned by the database query, and displays the landscape to the user. These steps are further defined with examples shown in Figures 7, 8, and 10, respectively.

Figure 7 is a flowchart showing example main steps used to receive and resolve a database query, such as step 610 in Figure 6. In Figure 7, step 612 receives the data query from a user 270. The query can be in any form translatable to the data repository in which the desired data is stored. For instance, the query can be a standard SQL query, already formatted, or could be a selection of fields from an HTML page. Whatever the initial form of the inquiry, the step 612 translates it into one that can be used to query a database or data repository. Step 614 is the actual retrieval of the data from the data repository, such as retrieving the list of data 410 from the data retrieval system 210.

Steps 616 to 622 concern the identification of the appropriate aspect for the virtual presentation of data retrieved from the data repository. The particular purpose of these steps, and generally of the entire inventive method, is to present the most significant view to a user by adapting the aspects of the landscape to data and other user characteristics in order to ensure an immediate and intuitive understanding of the presented data.

The ultimate purpose of the aspect determination component of the invented method is to reach the highest context sensitive capability for the aspect determination by implementing a function to determine the best aspect. In other words, the proper aspect is a function of various factors a,b,c,d, or: $\text{aspect} = f(a,b,c,d)$, where some of the parameters a,b,c,d that could influence the resulting aspect are: (a) user characteristics; (b) selected data type; (c) selection criteria; and (d) data contents.

The result of the steps from 616 to 622 is the identification of the appropriate graphic elements contained in the repository of virtual landscape aspect parameters 250 and in order to construct the appropriate aspect in the step 640.

Step 616 commences preparing aspect identification by identifying parameters concerning the user characteristics (a, above) and the selected data type (b, above) that will be used in the following step. An embodiment of the invention that can perform this step could be a server application able to manage the queries of multiple users on multiple database that are completely unrelated. The users could be unregistered net surfers, registered users, or customers, for example. For example, the embodiment on a single-server manages multiple data retrieval systems 210 (Figure 2) at the same time, like the results of a web-search using a typical search engine (Google, for instance). The different data retrieval systems 210 may contain, for example, a bookstore catalog, a holiday catalog, the presentation of a company, the catalog of an e-commerce application, and many others.

In order to ensure different presentations, an embodiment of step 616 detects all the parameters a and b concerning the user characteristics and the selected data type. For example, the parameters could be:

- identification of a data retrieval system to which the user will be connected (from the actual request, or a link selected by a user, or a software link protocol, e.g.);
- line connection speed (from connection protocol or from a user database, if the user is a registered user, e.g.)
- user preferences (from a user database, if the user is registered, or from a cookie resident on the client machine, e.g.)
- user retrieval rights (from a user database, if the user is registered, or from access rights tables belonging to the different data retrieval systems, e.g.). This parameter could see redundant with the access rights to the retrieval system. In this embodiment, the correct structure of the retrieval system is translated into significant effect, if any, on the visual characteristics of the virtual landscape. In an example embodiment, the retrieval system "bookstore catalog" allows the visitor to see the cover of the books but not to inquire about their contents. In this case, the embodiment will present to this user the inside of the store with the bookshelves appearing closed by glass windows. The user may look at the book covers, but not "touch" them in order to view their contents, as in the real world. For another user with different privileges, for

example a registered user, the bookshelves would be open and the customer allowed to browse the content of the books in addition to seeing their covers.

In the following step 618, these characteristics are translated into an array of codes, which belong to tables. These codes will be used in the following steps
5 to complete the aspect identification parameters. For example, the connection with a user could give (as parameters a and b) the following array of codes:

- data retrieval system:	HOL306	Holiday offer catalog of Pleasure Travel, Inc.
- line connection speed:	03	Medium Speed 22 to 128 kbps
10 - user preferences:	1742	Seashore environment
- user region:	DE	Germany
- user retrieval rights:	114	Only detailed denied

Step 620 performs the recognition of the parameter (c) (selection criteria)
15 and (d) data contents. This step is illustrated in more detail in Figure 8, which shows how the data selection criteria and data contents are translated into environmental characteristics

Figure 8 includes a multi-lingual dictionary of words and concepts 6202
and a table that stores concepts and environmental arrays 6204. These tables 6202 and
20 6204 are used by steps 6201, 6203 and 6205 to produce environmental characteristics, as described below. The tables can be stored on an appropriate server.

The first step 6201 is the recognition of the "significance" of the data or
of the criteria, and can be performed with the use of the table 6202, thus ensuring the
bridge between the concept hidden in the criteria and the data (words), and the
25 appropriate aspect belonging to the concept. In an embodiment of this step, the words contained in the criteria and the corresponding field of the retrieved data are related to the "environmental concept." The table 6202 (multilingual) has pointers from a plurality of different words to a single "environmental concept." For example the words "boat", "bateau", "sail", "vela", and "catamaran" (which name or describe boats
30 in several different languages) all point to the environmental concept of BOAT.

In this embodiment, the words contained in the criteria are corrected depending on the contents of the corresponding field in the rows found by the query (step 614) in the data retrieval system. For example, a query for "holiday on sail in

Crete" on a holiday database won't contain rows specifying "boat" as "accommodation form" and the concept "BOAT", and thus would not be used in the aspect determination. In this case only "Crete" would have influence for the aspect determination and the resulting aspect could become a seashore presenting hotels. In an
5 alternative case, if a query for "holiday on sail" would give results located only in Greece, also the "location" has an influence for the aspect determination and the resulting aspect would be a harbor with Greek background and blue water.

In the next step 6203, the "environmental concept" found for each word contained in the criteria or contained in the selected data are translated into an
10 "environmental characteristic." This happens by searching the most appropriate environment for a given "environmental concept", which are stored in the table 6204.

The description of the preferred embodiment contained in the table 6204 consists of a group of a given (potentially unlimited) number of environmental characteristics, in the following description called "environmental array."

15 In an embodiment of the step 620, the environmental characteristics consist of a three-dimensional array containing season, landscape-type and region. The array could be completely or partially filled, depending on the precision of the starting environmental concept. For example, the concept FISH would identify only the landscape type = "water surface" without any region or season. In another case, the
20 concept "RIMINI" would identify the landscape type = "beach", season = "summer" and region = "Italy".

In this embodiment the entire "environmental concept" to which the words contained in the data content criteria could point to differently filled "environmental arrays."

25 The next step 6205 chooses the best "environmental array" for all the concepts identified from the step 6201. In the embodiment of step 6205 conventional optimization techniques can be used, coupled with FIFO (first in first out) methods.

For example, the search for "Vacanza in Pedalo a Riccione" (a query in Italian) would give in step 6201 the following concepts:

30 Vacanza ----> Holiday
 Pedalo ----> Boat
 Riccione ----> Rimini

In step 6203, these concepts are related to their "environmental arrays" as follows:

"Holiday":

landscape type= "beach", season= "---" and region = "---".

5

"Boat":

landscape type= "water-surface, season= "---" and region= "---".

"Rimini":

landscape type= "beach", season= "summer" and region= "Italy".

This resulting best corresponding array chosen from step 6205 would be
10 landscape type = "beach", season = "summer", and region = "Italy".

Finally, the step 622 (Figure 7) assembles all of the collected parameters a, b, c, d, concerning the criteria and the data contents under the form of the "environmental array."

Using the former examples, the assembled result is an "extended
15 environmental array" containing the following codes:

- data retrieval system:	HOL306	Holiday offer catalog of Pleasure Travel, Inc.
- line connection speed:	03	Medium Speed 22 to 128 kbps
- user preferences:	1742	Seashore environment
20 - user region:	DE	Germany
- user retrieval rights:	114	Only detailed denied
- landscape type	8052	Beach
- landscape season	SU	Summer
- landscape reason	IT	Italy

25

This finally completed array of aspect parameters will be used by step 642 (Figure 9) to search in a central identification table of the repository of virtual landscape parameters 250 for the best corresponding aspect identity key. The key of this table has the same structure as the previously mentioned "extended environmental
30 array."

In the previous example, the search for the best corresponding aspect would give: Search key "HOL306 03 1742 DE 114 8052 SU IT" to yield an aspect id of #0002450. With this structure, the entire embodiment can offer different virtual

aspects depending of the change of a single component in the “extended environmental array.” For example, two different user databases, but having the same user/context/criteria situation could give aspect ids of:

5 Search key “HOL306 03 1742 DE 114 8052 SU IT” gives #0002450
 Search key “HOT121 03 1742 DE 114 8052 SU IT” gives #0008330
 Or, the same user database and same context/criteria could give aspect
ids of:

 Search key “HOL306 03 1742 DE 114 8052 SU IT” gives #0002450
 Search key “HOL306 04 3315 UK 114 8052 SU IT” gives #0019776
10 Or, the same user database and same user but different context/criteria
could give aspect ids of:

 Search key “HOL306 03 1742 DE 114 8052 SU IT” gives #0002450
 Search key “HOL306 04 3315 UK 114 8052 SU CH” gives #0109442
 In the same way that step 6203 can deliver partially filled keys, also an
15 embodiment of step 642 manages differently filled “extended environmental arrays.”
For example, the completely filled “extended environmental array” “HOL306 03 1742
DE 114 8052 SU IT” isn’t contained in the central identification table, and consequently
won’t give any aspect id. However, if the nearest referenced combination is “HOL306
03 ---- -- 999 8052 SU IT”, the aspect id #0708330 could be given. In absence of the
20 previous row, the selected combination “----- -- ---- -- --- 8052 SU IT” would give the
aspect id of #1000990.

 In the embodiment of step 642, conventional optimization techniques
used to recognize the best corresponding key could also be used to increase
performance.

25 The aspect identity keys found from previous steps are used in step 644
to search all graphical components (VRML data, textures, pictures, form parameters,
colors, positioning instructions, etc.) in tables and directories of the repository 250,
which are dedicated to computer graphic components. This graphic part of the
repository 250 is organized like a cascade of dependent tables where the dominant key
30 is the aspect identity key. Data structures and contents are created and populated with
standard tools known in industry, for example using Oracle database and VRML object
repositories.

Figure 9 is a flowchart showing example main steps used to construct an appropriate landscape, such as in step 640 in Figure 6. In Figure 9, step 642 receives and parses the aspect parameters assembled in step 622 of Figure 6. In step 644, these parameters are examined to determine which of the parameters can possibly correlate to a visual aspect of the landscape to be created. In doing so, the assembled parameters are checked within a table of the database, or with another database. For example, one of the aspect parameters sent in step 622 could be the above mentioned array:

	- data retrieval system:	HOL306	Holiday offer catalog of Pleasure Travel, Inc.
10	- line connection speed:	03	Medium Speed 22 to 128 kbps
	- user preferences:	1742	Seashore environment
	- user region:	DE	Germany
	- user retrieval rights:	114	Only detailed denied
	- landscape type	8052	Beach
15	- landscape season	SU	Summer
	- landscape reason	IT	Italy

giving as aspect search key the string "HOL306 03 1742 DE 114 8052 SU IT", (or "CUB" in a more simplified string shown in the Figures), and that parameter is searched in a table that stores some or all of the aspect factors. Figure 10 shows an example table 920 stored, for instance, in the repository of virtual landscape parameters 250. The table 920 is indexed by an "aspect ID". In this example, the aspect id relates to the physical shape of what will later be rendered in the landscape. In the first line of the table 920, the aspect Id "CUB" is associated with an aspect parameter 1 of a cube shape. Its aspect parameter 2 shows it associated with a grey color. Any number of aspect parameters could be assigned to "CUB" by appropriately populating the table 920, or other related tables.

The table 920 is used by the data representation system 200 to select how to show portions of the landscape to the user. For example, in a dataset 910, which is the results from a database query, all of the "types" returned are "CUB". In step 642 of Figure 9, one of the parameters of the appropriate aspect is "CUB". In step 644, the parameter "CUB" is identified as a visual characteristic of the landscape by searching in the graphical repository 250 all the graphical data that depend from identified aspect

“CUB” in the simplest embodiment, or from the aspect id #0002450 resulting from the extended environmental array “HOL306 03 1742 DE 114 8052 SU IT” of the more advanced embodiment of the previous example. In step 646, the table 920 is searched to see if any aspect parameters exist for the aspect Id “CUB”, and the different aspect
5 parameters 1, 2, ... n are retrieved. The data representation system 200 uses these parameters, as well as others, to build a two-dimensional landscape in step 648. In doing so, these aspect parameters are used to generate the data objects that will be placed in the two-dimensional landscape. Other factors, such as those used to provide the relative dimension of the x, y projection of the singular virtual object in order to fit
10 them contiguously in the two-dimensional maps are also factored in here. This can be specifically done by filling a matrix containing all the key-references of the 3D objects with the relative positioning and dimensioning information. Later, this two-dimensional landscape will be changed to three-dimensions in the step 670, and shown to the user
270.

15 In the table 920, associated with the aspect Id are parameter 1 (in this instance, shape) and parameter 2 (in this instance, color). These parameters are associated to the other data found in the dataset 910 and are used in creating a three-dimensional landscape 912, also shown in Figure 10. The landscape 912 contains several “buildings”, each of which are related to an item in the dataset 910. Because the
20 “type” in the dataset 910 was “CUB”, and the aspect parameter 1 for “CUB”, shown in table 920 is a cube shape, the buildings in the landscape 912 are formed with a cube shape.

Conversely, a dataset 930 in Figure 10 relates to a type of “CIL”. When the table 920 is consulted to help determine a landscape 932 for this dataset, the aspect
25 parameter 1 indicates a cylinder shape. Therefore, the cylinder shape is chosen to represent the types “CIL” in the corresponding landscape 932, when “CIL” types are present in the dataset 912. Note also that the displayed cylinders are white, which corresponds with the second aspect parameter for “CIL” stored in the table 920.

After the general landscape environment is determined in step 646, step
30 648 generates the single objects to be placed in the landscapes 912 and 932. For instance, the individual cubes are populated with the data 001, 002 and 003 for the landscape 912, and the individual cylinders are populated with the data 001, 002 and

003 in the landscape 932. Final assembly, in step 650, involves the above mentioned matrix containing the key-references of the 3D objects (stored using prior art techniques such as VRML or others) pointed by references in the matrix.

Once the landscape is assembled in step 650, the virtual landscape is rendered and displayed to the user 270 in the step 670 of Figure 6. An example flow for displaying a virtual landscape is shown in Figure 11. It is important to remark that all the following steps 672, 674, 676, 678 and 680 use prior art techniques such as VRML display functions, topological computation, etc., and do not singularly represent the inventive features of the present invention. In step 672, the display preparation facility 260 of the data representation system 200 receives the landscape parameters and virtual aspect parameters from the aspect construction facilitator 240. Step 674 computes the three-dimensional landscape from the received data, such as, for example, by establishing the point of view in the z-axis eventually to be shown to the user, and by establishing the height of the objects and backgrounds, etc. Additionally, forms, colors, appearance, and textures are all aspects that can be retrieved from the aspect parameter table 920 and used to create the three-dimensional landscape. This level of customization occurs all the way down to each object that is to be displayed to the user. Generally, having more aspect parameters stored in the aspect parameter table 920 allows the three-dimensional landscape to be created with more realism, and correspondingly better allows the user 270 to immediately understand the data presented. Figure 12 is a diagram showing differences in granularity in two aspect parameter tables 950, 960, and example three-dimensional landscapes able to be created from them, respectively, 952, 962. In the table 950, only three landscape parameters, form, color, and alignment are present in the table. Correspondingly, when the landscape 952 is created, the data representation system 200 can only produce a simple landscape, because there are no other parameters stored in the table 950 that allow differentiation in the displayed objects. In the landscape 952, all of the objects are grey cylinders that are aligned to the left. In contrast, the aspect parameter table 960 contains four aspect identifiers (Type, location, user nationality and shop) and six landscape parameters (Form, height, front color, roof color, first alignment, and second alignment). When used together, the data representation system 200 is able to produce a landscape 962 that can convey much more information by the many different variables

than can be produced in the landscape 952. In the landscape 962, several of the landscape parameters can differ from one shop to another. For instance, because one of the landscape parameters stored in the table 960 is height, then the objects shown in the landscape 962 could have different heights. There is a landscape parameter for "front color" in the table 960, and there are two different types of front colors for the buildings shown in the landscape 962. The same is true for roof colors. Additionally, alignment variables may be stored in the table 960. Variables for objects that will appear in the three-dimensional landscape but are not tied to any data entry can also be stored in the table 960. For example, the eventual landscape may be a dock where the data elements are shown as different boats. Coordinates of the water in which the boats are setting may be stored in the table 960, and the data representation system 200 would then know not to populate the area where the water will be shown. Storing all of this data in the aspect parameter tables gives the data representation system 200 the ability to present the data in the three dimensional landscape presentation with as much realism as desired, still using the principles of the invention.

The three-dimensional landscape view, such as the landscapes 952 and 962 of Figure 12 are generated in step 676 based on the outcome of step 674. This step specifically involves, for example, interfacing to conventional 3D graphic cards, to DirectX software by Microsoft, or to Java applets emulating the functions of a graphic card, any or all of which can be running on the client display device. Data that will be displayed with the image created in step 676, for instance, the data 001, 002 and 003 in the landscape 912 of Figure 10 is populated into the landscape in step 678. In generating the three-dimensional landscape, factors such as the user characteristics, selected data type, selection criteria and data contents are all used to create the three-dimensional landscape to be shown to the user 270.

Finally, after the landscape has been created and the data populated within it, the landscape image is displayed to the user 270 in step 680. Generally, the image is displayed on a standard computer screen, but could be in any form of image display, such as on a portable device using a wireless connection, for instance on a Personal Information Manager, cellular phone, or portable computer using a wireless data connection.

Once the landscape is presented to the user 270, the user can use standard input devices to control a three-dimensional movement within the landscape. For instance, the user could use a keyboard or mouse to "walk" down a road, passing some of the data items that have been presented in the landscape and revealing others as the display has room for them. Additionally, the user can turn directions, and go up or down if the area of the landscape allows it. For example, the user may be able to navigate to a set of steps that separates a first level of shops from a second level. By moving the input device appropriately, the user could climb the stairs to interact with the shops presented on the second level. Movement within three-dimensional landscapes is well known, and those skilled in the art could implement such movement without undue experimentation.

In one such example, the landscape presented could be one of a group of sailboats in a harbor, based on a search of sailboats available to be rented for a given time. The color of the water could be blue to indicate that the selected location was in the Mediterranean, or green if the location was in the Caribbean. Houses around the harbor could be colored blue and white to indicate that the only sailboats that matched the query were Greek. If the desired month to check for availability was March, the sky could be cloudy, to reflect the typical weather in Greece in March. Each sailboat presented could have a "plaque" on the deck giving the boat's name, price, and other data. To view this data, the user moves near the plaque, where the data is visible on the screen. To look at the other boats that came up in the search, the user walks along the pier to the next boat, allowing the user to see the data on the other plaques.

Additionally, landscapes within landscapes are possible. For example, in the sailboat example above, the user could navigate inside a selected one of the sailboats. Once inside, the landscape would change to provide additional detail about the boat itself. For instance, the landscape could show 3 bedrooms to indicate that there are actually three rooms in the selected boat, and the relative size could be shown. Furniture included with the boat could be represented. The number of bathrooms could be easily represented by showing them and their locations. Additionally, the specifics about number and size of rooms, etc., could be listed in data form in a special text box that appears within the display, which could be used instead of, or in addition to the graphical representations. This embodiment is a hybrid example of data presented in its

traditional form, such as in lists or tables, and in graphic form as described herein. In other embodiments the text boxes would not necessarily be displayed.

Figures 13-16 show additional examples of how a query to a database that would otherwise have been returned in text form can be returned in a three dimensional graphic form that can be easily interpreted.

Figure 13 includes a sample database listing 130 of items in a holiday database. Included are different forms of accommodation, which include rooms, chalets, bungalows, and even sailboats and campers. Listed with each type of accommodation is the type of holiday linked therewith. The holiday could be spent in towns, in the country, on the beach, on a sailboat, or in other forms.

In the previous examples, the landscape based on a data query was based on at least the user characteristics, selected data type, selection criteria, and data contents. In the following examples, the selected data type has been simplified because it is always the same – a holiday catalog. In this case, therefore, some of the characteristics that can be used to create the three-dimensional landscape are holiday location, accommodation form, and season. The characteristics used to determine the landscape would be stored in or accessed by the aspect construction facilitator 240 of Figure 2, and used in the method described above with reference to Figures 6-9 and 11.

The database listing 130 of Figure 13 shows a convention representation of an example holiday database. Holiday opportunities are characterized by an accommodation form, a purpose, a location, etc. If a query were performed on the listing 130 having “room” as the accommodation form and “town” as the location type, the corresponding subset 133 is created.

Figure 14A is a front view of a three-dimensional virtual landscape 136 generated from a Java Applet as an answer to the same query that generated the subset 132 of Figure 13, but by using the data representation system 200 described above. A Java Applet is a Java program running in a Web browser window of a personal computer. The landscape 136 is adapted to a downtown location, to denote the location type in the query. The landscape 136 includes eight town-buildings corresponding to the eight town-hotels that were selected in the subset 132. Figures 14B, 14C and 14D show the data manipulation effect of navigation through the landscape 136, performed by the user “walking” in the roads of the virtual town created by the query that

produced the subset 132. The landscape 136 is populated by objects that directly correspond to the selected data. Consequently, moving within the landscape 136 shows how other data from the subset 132 can be viewed by the user. The figures 14B, 14C and 14D represent three moments of a continuous rotation performed by the user, during a "turn right" request. As seen in Figure 14D, the road to the right of the building is empty because the subset 132 only contains eight object that do not need to be further sub-grouped into categories. If there were more types of categories selected in the subset 132, additional buildings could be placed on other streets, each street denoting a unique type of category.

Figure 15 shows another three-dimensional landscape 156 created from a subset 152 of the database listing 130. The subset 132 was created with a query that selected a "room" type of accommodation in a "country" location from the database listing 130. In the landscape 156, the four holiday offers are presented as four buildings in a green countryside.

Figure 16 shows a completely different three-dimensional landscape 166 created from a subset 162 of another database listing (not shown). The landscape 166 shows the inside of a library, where the database concerned a shop selling products. To create the subset 162, the query used had a product type of "book" and a product category of "science." In the landscape 166, the six populated shelves (similar to the different "roads" in Figures 14 and 15) contain books relating to the six "product subcategories" contained in the subset 162. By turning onto one of the shelves, the aspect of the chosen shelf will appropriately change in order to meet the requirement of the aspect parameters configured for the criteria component sub-category.

Each of these examples shown in Figures 13-16 can be performed with the data representation system 200 shown in Figure 2, by varying which factors are stored in the repository of virtual landscape parameters 250, and the relationship between them and the query results as determined by the aspect construction facilitator 240.

The data representation system 200 can be used to synthesize virtual shopping centers for e-commerce on the Internet. The virtual shopping center is animated, easy to retrieve information from, and is controllable by the user. In operation, an inquiry in a database containing shops grouped in one of multiple different

shopping centers could be presented differently for each shopping center. In the same manner, an inquiry in a database containing products sold by multiple different shops can be presented as different interiors of each shop. In the first case, the aspect parameters used by the aspect construction facilitator 240 would contain parameters
5 bound to the type of data (shop) and the selection criteria (a particular shopping center) to create a virtual view of shops within the particular shopping center. In the second case, the aspect parameters used by the aspect construction facilitator 240 would contain parameters bound to the type of data (product) and the selection criteria (a particular store name) to create the virtual inside of that particular store. Another
10 embodiment could still show the particular shopping center of the first example, but show just the shops containing the products desired, as a type of filter. Any of these implementations, and many more that spring from them can be easily executed with the data representation system 200 described herein.

Changes can be made to the invention in light of the above detailed
15 description. In general, in the following claims, the terms used should not be construed to limit the invention to the specific embodiments disclosed in the specification and the claims, but should be construed to include all methods and devices that are in accordance with the claims. Accordingly, the invention is not limited by the disclosure, but instead its scope is to be determined by the following claims.